2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), Benghazi, Libya 21-23 May 2023.

2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA) | 979-8-3503-1989-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/MI-STA57575.2023.10169409

Calamity classification in satellite and drone images using CNN and Transfer Learning

Seddik KHEMAISSIA
Laboratoire de Génie Electrique
Larbi Tebessi University – Algeria
seddikk@hotmail.com

Youcef SOUFI
Laboratoire de Génie Electrique
Larbi Tebessi University- Algeria
y_soufi@yahoo.fr

***Abstract*— Every year, natural disasters strike, wreaking havoc on lives, property, and the environment forever. They are unstoppable phenomena. In recent years, remote sensing imaging analysis has become increasingly important for the purpose of identifying and monitoring natural disasters in the context of climate change and environmental surveillance. The ability to convey a large amount of information in a single piece while still capturing the status of the underlying ground is one advantage of aerial or satellite imaging. Remotely sensed imagery has become increasingly important for environmental and climate monitoring in recent years, particularly for the detection and management of natural disasters. Using a convolutional neural network (CNN) to better extract the catastrophe characteristic, we propose autonomous natural disaster identification in this paper. CNN is resistant to shadows and is able to accurately determine the characteristics of a disaster. It is also able to avoid operator error, which can have an impact on the efficiency of disaster relief efforts. The images used as the model's training data are divided into four categories: floods, cyclones, wildfires, and earthquakes. The input for disaster detection is either a live video stream or a video that has already been recorded. The study demonstrated that transfer learning can be used to accurately recognize natural disasters. We hope that the findings of this study will lead to the creation of monitoring or surveillance systems that are capable of accurately recognizing natural disasters in real-time. Based on the encouraging results, the suggested method may contribute to our understanding of deep learning's role in catastrophe detection.**

Keywords— Catastrophe detection; airborne image; transfer learning; CNN.

I. INTRODUCTION

Natural disasters frequently arise in ways that are difficult to predict. They are exacerbated by the actions of other people and defy human will. Due to their size, complexity, frequency, and economic impact, natural disasters are also becoming increasingly common. Every nation is obligated to protect its inhabitants, infrastructure, and other heritage assets from environmental and natural disasters.

Similar to military defenses, civil defense employs a variety of measures to lessen the dangers posed by "aggressions," such as earthquakes, inundations, terrain slippage, drought, and environmental disasters. The ultimate goal of surveillance is to evaluate risks as accurately as possible, regardless of their origin. All surveillance is the result of observations that are as precise and persistent as possible and of "Models" that enable their interpretation. The goal of focusing on "sensitive zones" was to find out when a dangerous phenomenon first appeared and how it progressed. [1]

Automation offers a fascinating alternative. The ability of machines to interpret terrestrial images would make it possible to multiply observations in order to obtain both information and models. The current state of the art in machine learning relies heavily on deep neural networks, whose abilities in image classification, object detection, and shape recognition have enabled significant advancements in artificial intelligence. In order to automate the interpretation of satellite and airborne images, we aim to develop, implement, and validate deep neural network models.[2]

Machine learning-based detection has been the subject of a significant amount of research. [3] proposed employing hierarchical form characteristics in the bags-of-visual-words scenario to detect large-scale damage. [4] shows how to forecast cyclone paths using artificial neural networks. [5] investigates how successfully Random Forests, radial basis functions, and multilayer neural networks handle earthquake damage in 2010. Later,[6] utilize the scene's 2D and 3D features to locate damage. Ying Liu [7] uses deep learning algorithms to recognize geological disasters at the same time. CNN [8] is one of the most well-known deep learning methods for improving the detection of an earthquake, tsunami, or eruption-related results.

Thus, the intersection of automatic learning, computerized vision, and remote detection is the focus of this article. Particularly, we suggest utilizing techniques based on deep learning that is capable of delivering an alarm signal and detecting catastrophes in real time [9]. The detection of natural disasters can be automated using these techniques. We developed a CNN model as a response, and we estimate the confusion matrix to see how well it performs. In addition, the model is tested using pre-registered satellite and drone videos. The experimental results are precise and effective when tested using YouTube videos.

The sections of our paper are as follows: We begin by outlining the humanitarian and emergency context (section 1) and presenting related works as the foundation for our method (Section 2). The dataset and the experimental setup are all discussed in the following section. After that, we go over the results of our computational experiments (Section 3) and suggest a new incident workflow. In conclusion, we offer some

979-8-3503-1989-7/23/$31.00 ©2023 IEEE

introductory remarks and open the discussion to other works (Section 4).

II. Convolutional neural networks (CNNs)

CNN is a form of deep learning model used in segmentation and other machine vision applications. When compared to other network types, convolutional operations greatly reduce the number of hyperparameters in the network, making learning more efficient. The majority of CNNs are made up of several convolutional and pooling layers, followed by a fully-connected layer that provides network output. Convolutional layers are made up of a series of filters that form three-dimensional structures. For an RGB image, the first convolutional layer may contain a filter with a size of 7x7x3, which implies it is 7 pixels wide and high and covers all three input channels. Convolution is achieved by sliding the filter over through the image pixels and calculating the dot-wise product of the input values and the filter weights. Pooling layers are widely used between convolutional layers to reduce the input to the next layer. By lowering the number of parameters required, deeper network designs can be used. The most popular pooling operation for downsampling is maxpooling. It finds the maximum input value for each filter value and outputs it. Fully-connected layers are typically used as output layers since they have connections between the neurons in the fully-connected layer and all of the previous layer's outputs. (Fig.1)

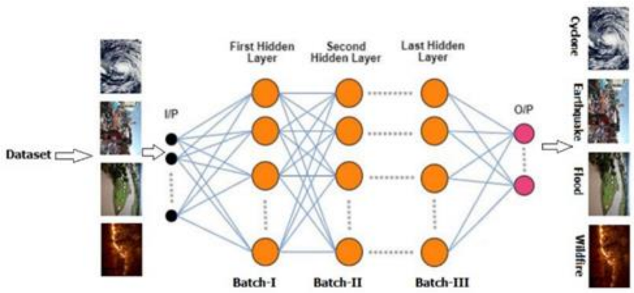

Fig. 1. Disaster classification using CNN

A) Convolutional Layer

A convolution layer is a crucial part of the CNN model that enables feature extraction and is often composed of a combination of linear and nonlinear operations, namely the convolution function and the activation function. Convolutional Layers are composed of filters that are convolved on image pixels in order to extract features.

In general, the size of those features (width and height) is small when compared to the input data. The resulting feature, however, could be identical to the source images if the border of the image pixels is padded before the convolution procedure. Simple convolution refers to one-dimensional signal processing convolution. When a convolution action is done between two signals that span along two dimensions that are mutually orthogonal to one another, the process is known as 2D convolution. CNN is easily adaptable to operate with input signals of varying sizes. Based on the problem domain, it may be necessary to apply zero-padding to the input feature before performing a convolution operation. The technique of surrounding a matrix using zeroes is referred to as zero-padding in convolutional neural networks. This can contribute to the preservation of features at the boundaries of the original matrix as well as regulating the size of the resulting feature map.

Convolutional Layers are often utilized in neural network image processing. The computation of an image through a convolution layer achieves the same impact as putting an image filter. Moreover, the weight of the edge defines the filter's characteristics. In other terms, the extraction of features is the responsibility of the convolution layer.

B) Pooling Layer

To ensure translational invariance, a Pooling Layer subsamples the output of the lower layer. It, like the Convolution Layer, has a direct connection to the lower layer (Fig. 2). However, the procedure of obtaining the node's value differs from that of the convolution and fully-connected layers. The result produced by the local node in the bottom layer is propagated to the top layer.

The pooling Layer is also often used in neural network image processing. The calculation of an image has no effect on its value, specifically after minor modifications, since only the highest values in a given area is taken and brought to the top layer. As a result, even with minor changes within the range, the pooling layer returns the same value. In other terms, even if the image changes little, the effect remains the same. Notably, none of the pooling layers have parameters that can be learned, despite the fact that filter size, padding, and stride are hyperparameters in pooling processes, similar to convolutional operations..

C) Fully-Connected Layer

The final pooling layer's output feature maps are frequently flattened into a one-dimensional (1D) array before being transmitted to the network's final outputs, such as the probabilities for each class in classification tasks. The majority of the time, the number of classes in the final fully connected layer is the same as the number of output nodes. After every fully connected layer, a nonlinear function like ReLU is used.

D) activation function

The final dense layer's activation function is frequently distinct from the rest. A softmax function is used to normalize output real values from the dense layer to target class probabilities in the multiclass classification task, where each value ranges from zero to one with a sum of one.

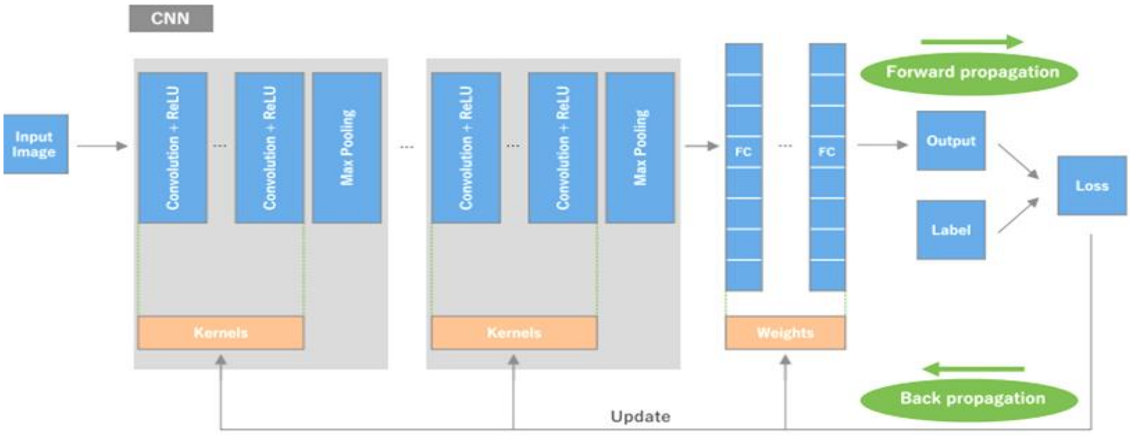

Fig.2. Convolution neural network architecture and the training process

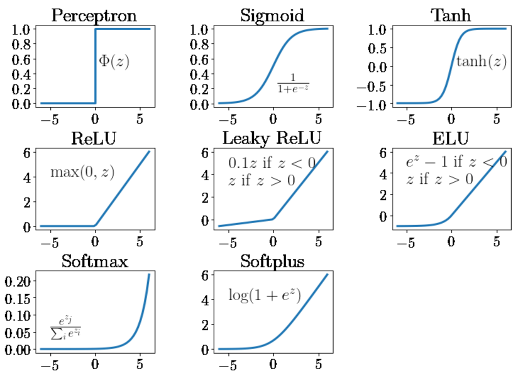

Fig.3. Activation Function

E) *Training a network*

When fitting a neural network, a training dataset is used to update the model weights and get a good mapping of inputs to outputs. An optimization algorithm that looks through a set of possible values for the neural network model in order to achieve satisfactory results on the training data is used to solve this training procedure. The stochastic gradient descent method, which makes use of error backpropagation to update model weights each iteration, is the most suitable general strategy for dealing with this issue. (Fig. 2).

F) *Loss function*

The consistency between the provided dataset labels and the network's onward propagating output predictions is evaluated using a loss function, also referred to as a cost function. For multi-class classification, cross-entropy is a popular loss function, but for regression to binary variables, the mean squared error is commonly used. One of the hyperparameters is the type of loss function, which must be determined based on the tasks.

G) *Gradient descent*

Gradient descent is a popular optimization technique that reduces loss by iteratively adjusting the learnable parameters of the network, such as kernels and weights. Each learnable parameter is updated with an arbitrary step size determined by the learning rate in the negative direction of the loss function gradient, which represents the direction in which the function has the steepest rate of increase (Fig. 2).

In practice, the loss function's gradients are computed using a mini-batch subset of the training dataset for parameter updates due to memory constraints. Mini-batch gradient descent is both a hyperparameter and a stochastic gradient descent (SGD). In addition, a number of enhancements to the gradient descent algorithm have been proposed and are widely used, such as SGD with momentum, RMSprop, and Adam.

H) *Transfer learning*

It is very difficult and time-consuming to collect images from a particular area of interest and train a classification from scratch. We start with a pre-trained model and modify the final few layers before classifying images. A pre-trained network can be used in two different ways in practice: extraction and fine-tuning of predefined features (Fig. 4). The inner layers remain unchanged from the pre-trained model, with the last layers being modified only to accommodate the number of classes. In this instance, we make use of the trained ResNet50..

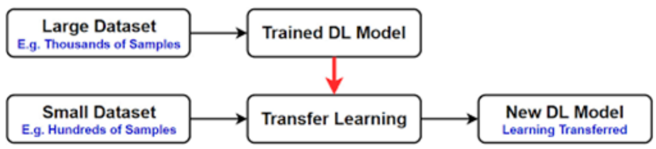

Fig. 4. Transfer Learning approach

I) *Data augmentation*

The accuracy of a deep learning model is significantly influenced by the quality, availability, and environmental significance of the training data. However, a lack of data is one of the most frequent difficulties encountered when developing deep learning algorithms. In industrial use cases, collecting such data can be costly and time-consuming. To develop high-precision AI systems more quickly and reduce reliance on the collection and processing of training samples, businesses use data augmentation, a low-cost and effective strategy.

Data augmentation is the process of producing additional sets of data from existing data to significantly increase the quantity of data. Examples of this include adjusting the data in a small way or using machine learning techniques to acquire new data points in the latent space of the source data to expand the dataset.

Synthetic data: Generative Adversarial Networks are typically used to generate artificial data without the use of genuine images..

Data augmentation: generated from real images by making minor geometric changes like flipping, translating, rotating, or adding noise to make the training set more diverse.

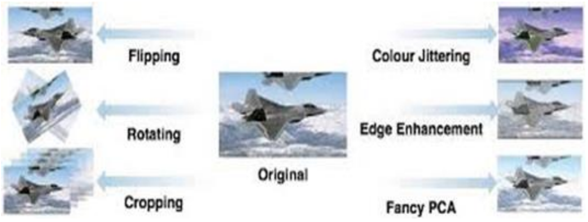

Fig. 5. Data augmentation is the technique

J) Dropout Regularization

A dropout is a regularization approach that avoids overfitting by assuring that no entities are dependent on one another. The process of disregarding specific nodes in a layer at random while learning is referred to as "dropout" in deep learning. In the figure below, the neural network on the left represents a standard neural network with all nodes activated. The red elements on the right have been removed from the system; their biases and weights are not considered throughout learning.

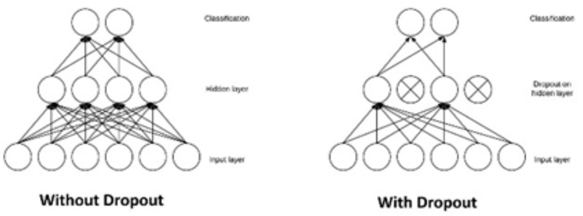

Fig. 6. Dropout Regularization

Dropout is still a widely common preventive mechanism over overfitting because of its efficacy and reliability, despite a plethora of alternatives such as Early stopping, Weight decay, Noise, and model combination..

III. Results and Discussion

In this section, we will investigate how Keras and its learning algorithm can be used to detect natural disasters in images and video streams using computer vision and deep learning methods. We'll attempt to comprehend the software and libraries utilized in the implementation before going on to practice. To resolve image classification issues fast and easily, we will employ an automatic learning technique known as transfer learning.

The suggested deep convolutional neural network was run on a computer with a Core i7, a CPU of 2.8 GHz, and 32 GB RAM Memory, and various types of results were computed.

A) Dataset and Preprocessing

The data used in our study was gathered from reader Gautam Kumar and who shared it on his LinkedIn profile. Gautam utilized Google Images to compile a collection of 4,428 images divided into four categories: The collection contained four classes: cyclone, earthquake, flood, and wildfire, as shown in Fig. 3, having 928, 1350, 1073, and 1077 images, respectively.

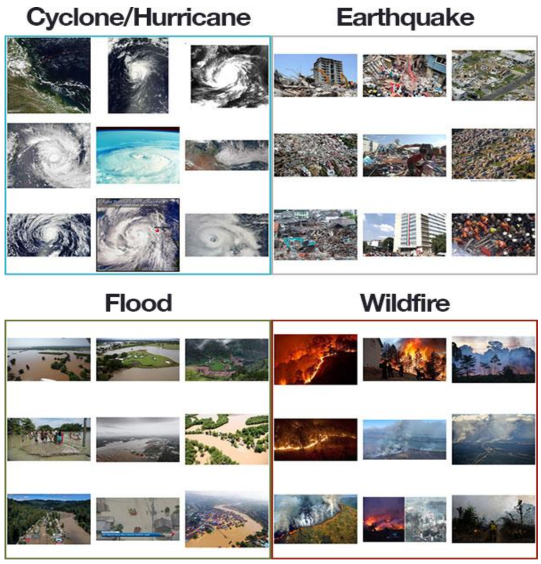

Figure 7. the 4 classes of natural disasters from the dataset.

An adaptive histogram equalizer was used during the reprocessing of the dataset to get rid of the noise. There were three categories for the entire dataset: testing, validation, and training. The dataset was used for testing with 23%, validation with 17%, and training with 60% of the data. By employing these percentages of the dataset, the values of the dataset that would be utilized for testing, training, and validation were communicated to the machine. The number of epochs during the training phase was counted with the help of the validation set. Using a previously trained CNN, we applied a transfer learning technique based on ImageNet weights. We used ResNet50 as a CNN model, and the GAP (global average pooling) layer and all other inner layers were left alone. Pooling, the dense layer, the Dropout layer, and the dense layer were activated (softmax) after the FC layer was removed and replaced by the medium layer.

B) Performance of the model.

the performance of a test image from the obtained data set is evaluated. The evaluation metric is measured, which comprises the system's precision, recall, and accuracy. The trade-off between false positives and true positives is also shown by plotting the ROC curve. The model's performance was evaluated by plotting the accuracy recall and AUC.

To determine the best system performance, three case studies will be conducted.

Case 1: Epochs: 80, images: 400, Resolution 80x80

The number of epochs varies, but the number of images and resolution remains constant.

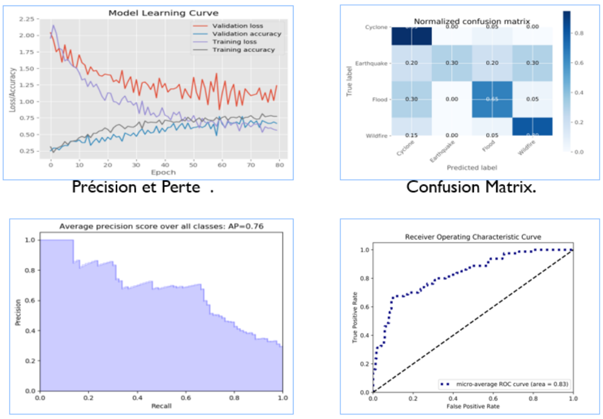

Fig. 8. Case 1: Performance

CASE 2: EPOCHS: 80, IMAGES: 400, RÉSOLUTION 224X224

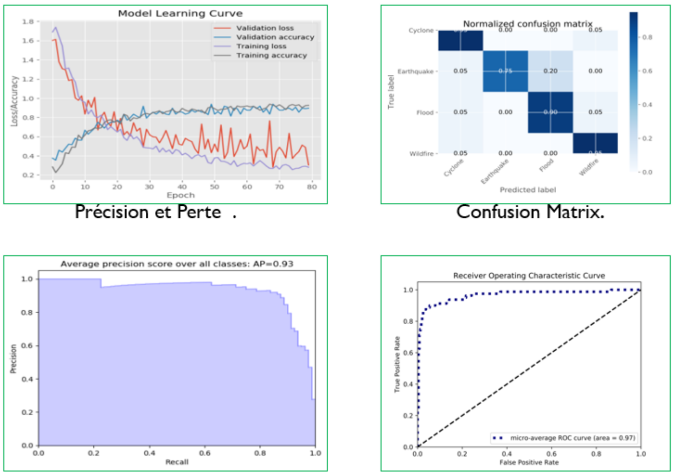

Fig. 9. Case 2: Performance

CASE 3: EPOCHS: 80, IMAGES: 800, RESOLUTION 224X224

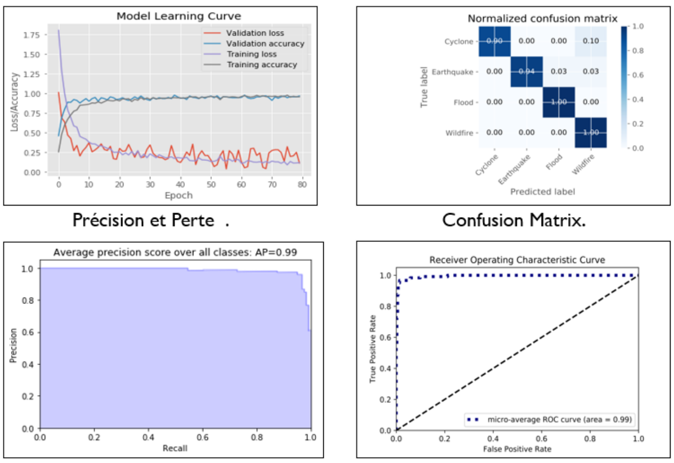

Fig. 9. Case 3: Performance

Following the examination of the obtained results, the following observations are made:

1- The precision of learning and validation increases with resolution, indicating that a higher resolution helps the model learn better.

2- The confusion matrix is superior. A better resolution provides more assurance to our learning system.

3- The same is true for the ROC, and the recall begins to improve as the system learns more. A better resolution increases the likelihood of a better classification.

IV. CONCLUSIONS

In the event of a disaster, the provided real-world disaster detection method has the ability to generate an alarm automatically. This mechanism can be put to two separate uses. Analyze satellite images using a satellite camera to anticipate and avert disaster. A polar satellite can easily capture a direct flux of forested areas, mountain ranges, rivers, and inaccessible areas, and if it finds a sign of disaster, the system can automatically generate an alert. A geostationary satellite can detect a cyclone above the coastal area. Additionally, it significantly slashed the time required for satellite image analysis and human effort for each disaster. In the event of an earthquake, the system can keep track of the specifics of the affected and destroyed areas. Additionally, the proposed approach enables us to rapidly notice and automatically make an alarm in the event of an incident, assisting us in preparing for these catastrophes. Consequently, the proposed system can assist the research community in anticipating and minimizing the damage caused by natural disasters and can be used in a manner analogous to an "il in the sky." The detection of additional natural disasters, such as terrain slides and volcanic eruptions, could benefit from this concept.

REFERENCES

[1] G. Giulani and P. Peduzzi, « The PREVIEW Global Risk Data Platform: a geoportal to serve and share global data on the risk to natural hazards », Natural Hazards Earth System Sciences, 11, pp. 53-66. 2011

[2] X. X. Zhu et al., « Deep Learning in Remote Sensing: A Review », arXiv preprint arXiv:1710.03959, 2017.

[3] L. Gueguen and R. Hamid, “Large-scale damage detection using satellite imagery,” 2015 IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1321–1328, 2015.

[4] R. Kovordányi and C. Roy, “Cyclone track forecasting based on satellite images using artificial neural networks,” ISPRS J. Photogramm. Remote Sens., vol. 64, pp. 513–521, 2009.

[5] A. J. Cooner et al., “Detection of Urban Damage Using Remote Sensing and Machine Learning Algorithms: Revisiting the 2010 Haiti Earthquake,” Remote Sens., no. Ml, 2016.

[6] J. Tu et al., “Automatic Building Damage Detection Method Using High-Resolution Remote Sensing Images and 3D Gis Model,” ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci., vol. III-8, pp. 43–50, 2016.

[7] Y. Liu and L. Wu, “Geological Disaster Recognition on Optical Remote Sensing Images Using Deep Learning,” Procedia Comput. Sci., vol. 91, pp. 566–575, 2016.

[8] M. B. Bejiga et al, "Convolutional neural networks for near real-time object detection from UAV imagery in avalanche search and rescue operations," IEEE Int. Geosci. Remote Sens. Symp., pp. 693–696, 2016.

[9] S.N.K.B. Amit, and Y. Aoki, "Disaster detection from aerial imagery with a convolutional neural network". In Proceedings of the International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES - KCIC), Surabaya, Indonesia, pp. 239–245, 2017